

Application No. 10/021,080
Response Dated 02/20/2006
Reply to Office Action of 11/01/2005

PATENT

Agent's Docket No. 11922-US

Remarks and Arguments:

In complying with the Examiner's requisition under 35 U.S.C. 101, Applicant elects to retain in the application claims directed to network management and service provisioning framework. Without prejudice, claim 11 has been cancelled and claims 1 to 10 have been recast to claim a network management and service provisioning framework for use in a network management and service provisioning system. In order to comply with the Examiner's requisition, certain terms have been amended only in format to ensure correct technical effect.

Independent method claim 12 was also amended to correspond to the amended independent claim 1 to ensure unity of invention. Dependent method claims 22 and 23 were consequently amended to ensure proper antecedents in view of amendments to independent method claim 12.

Consistently paragraphs [19a] and [19b] corresponding to amended independent claims 1 and 12 have been added to the Summary of the Invention section of the specification.

The "network management enabling technologies 230" in paragraph [29] has been amended to read --network management and service provisioning technologies 230--. Support for the amendment is found in independent claim 1 as originally filed.

It is submitted that no additional subject matter has been introduced by the amendment.

Consequently, Applicant respectfully requests that the amendments after final Office Action be entered as these amendments comply with the Examiner's suggestion to amend and do not raise new issues.

Application No. 10/021,080
Response Dated 02/20/2006
Reply to Office Action of 11/01/2005

PATENT
Agent's Docket No. 11922-US

On pages 3 to 5 of the 11/01/2005 Final Office Action, the Examiner states: "Applicant's arguments alleging Haggerty's failure to disclose 'run-time registration of at least one plug-in' have been fully considered but they are not persuasive." Without prejudice, the applicant takes this opportunity to respond to the Examiner's new objection to claims 1 to 3, 6 to 9, 12 to 20, and 22 to 24, under 35 U.S.C. 103(a), intending to address any outstanding issues arising from the previous 35 U.S.C 102 objection, and in particular to expand on the applicant's previous submissions.

The Examiner rejected claims 1 to 3, 6 to 9, 12 to 20, and 22 to 24, under 35 U.S.C. 103(a) as being obvious in view of Haggerty and Van de Vanter.

Amended claim 1 of the present application is directed to a framework which includes a plug-in registry configured to load and register at run-time an enabling technology plug-in which brokers access to a network management and service provisioning enabling technology. The registry is an element not taught by Haggerty. In order to establish a prima facie case of obviousness every claimed element must be found in a prior art reference. The Examiner must point out which prior art element is viewed to be equivalent to the plug-in registry.

The Examiner states that Haggerty's resources correspond to the network management and service provisioning enabling technologies claimed in the application. Applicant respectfully states that Haggerty's topology objects do not represent enabling technologies. The Examiner quotes the last paragraph in the second column on page 75 of Haggerty stating "Managed Objects, which represent network entities and resources." It is the Applicant's position that a person of ordinary skill in the art, upon reading Haggerty, would have understood "Managed objects, which represent network entities and [their] resources." Support for this

Application No. 10/021,080
Response Dated 02/20/2006
Reply to Office Action of 11/01/2005

PATENT

Agent's Docket No. 11922-US

interpretation is found in the Haggerty reference in the second paragraph of column 1 on page 76 "Topology objects represent the network and contain information pertaining to addressing, type, uniqueness, resources, and status" the combined logical interpretation of Haggerty's teachings would be: managed objects include topology objects which represent network entities, network entities which represent the network, topology objects which contain information pertaining to network entity addressing, network entity type, network entity uniqueness, network entity resources, and network entity status. Further support for this interpretation is found in Haggerty, particularly in Fig. 4 of Haggerty which corresponds to the derivation hierarchy 300 described in the present application. Fig. 4 of Haggerty does not show even a single network management and service provisioning (protocol) enabling technology. It is respectfully submitted that the network management and service provisioning enabling technologies claimed in the application are not resources in the sense described by Haggerty, and therefore the network management and service provisioning enabling technology plug-ins are not topology objects in the sense described by Haggerty.

The plug-ins of amended claim 1 of the present application are loaded and registered with the plug-in registry at run-time. As previously argued, Haggerty teaches away from loading and registering at run-time in footnote 1.

A person of ordinary skill in the art would understand that the CMIP-gateway of footnote 1 relates precisely to the kind of a network management and service provisioning enabling technology as clearly described in paragraphs [29] and [33] of the present application. The applicant respectfully submits that Haggerty's teachings in footnote 1 regarding the CMIP-gateway are particularly important and should not be overlooked in determining what Haggerty teaches. Haggerty teaches the hard-coding of the CMIP-gateway, a network management and

Application No. 10/021,080
Response Dated 02/20/2006
Reply to Office Action of 11/01/2005

PATENT

Agent's Docket No. 11922-US

service provisioning enabling technology, and linking thereof in a network management system.

This is relevant to Haggerty's treatment of enabling technologies and as this CMIP-gateway is hard-coded, Haggerty teaches away from the invention claimed in independent claim 1 which relates to run-time loading of a CMIP plug-in and the registration thereof with a plug-in registry.

The present application describes the plug-ins as shared library object code, and particularly compiled unlinked object code which is loaded at run-time (see paragraph [33] of the present application). Even if the topology objects of Haggerty could be equated with the plug-ins of amended claim 1, a person of ordinary skill in the art would not equate the topology object creation by auto discovery with the run-time loading and registration of an enabling technology plug-in object code (as recited in claim 1) because that would lead to the impossibility of equating the instantiation of an object type with the loading-up of compiled object code at run-time.

The plug-ins of claim 1 are loaded and registered with a plug-in registry. The applicant respectfully submits that Haggerty does not teach such a plug-in registry. The Examiner appears to have equated the plug-in registry with either OpenView Map additions or with the MOM. However, as the Examiner states, the OpenView Map component of Haggerty's system is a network management and service provisioning application, and is not responsible for registration of plug-ins. The MOM is not described as performing any actions in the passage cited by the Examiner. If the Examiner persists in stating that Haggerty teaches a plug-in registry to load and register plug-ins, as claimed in the present application, the examiner is kindly requested to clearly indicate where Haggerty teaches such a plug-in registry.

The framework of amended claim 1, of the present application also includes an executable code implementation of a single managed entity object class, the single managed

Application No. 10/021,080
Response Dated 02/20/2006
Reply to Office Action of 11/01/2005

PATENT

Agent's Docket No. 11922-US

entity object class being run-time derivable via type derivation into a derivation hierarchy of managed data network object types based on run-time parsed entity derivation directives, a separation being achieved between managed data network entities and enabling technologies, the separation enabling independent development, maintenance and troubleshooting of network management and service provisioning deployments minimizing the need to re-code and re-compile framework code in support of new managed entity object types. However, the Examiner equates the topology objects with both the run-time registrable enabling technology plug-ins and the derived managed object types. The applicant does not believe that the topology objects of Haggerty can be equated with either of the elements of claim 1, but even if they could in accordance with the Examiner's interpretation of Haggerty, joint representation of these claim elements by topology objects is clearly not a separation between managed data network entities and enabling technologies. The applicant therefore respectfully submits that Haggerty does not teach distinct enabling technology plug-ins and managed entity object types as recited in claim 1.

Amended independent claim 1 further limits the invention to a network management and service provisioning enabling technology plug-in having an associated run-time loadable lexical analyzer stub. The Examiner appears to agree with this position in the first paragraph of page 7 of the final Office Action. It is noted that the Examiner goes on to state "some form of parsing and lexical analysis would be necessary for [Haggerty's] system to import, convert, and understand the added plug-ins". However, as stated above at paragraph [33] of the specification, the plug-ins include compiled object code. The object code contains a sequence of instructions executable by a processor. No parsing and lexical analysis is required in practicing the present invention in loading and registering the plug-ins with the plug-in registry. Furthermore, parsing and lexical analysis steps performed in loading and registering the plug-ins with the plug-in

Application No. 10/021,080
Response Dated 02/20/2006
Reply to Office Action of 11/01/2005

PATENT

Agent's Docket No. 11922-US

registry are neither described nor claimed. In accordance with the claimed invention, no plug-in conversion and understanding of the loaded and registered plug-ins are required.

The framework of amended claim 1 includes a directive parser configured to process, at run-time, a self-contained managed data network entity specification file including directives. The claimed invention achieves separation between managed entities and network management and service provisioning enabling technologies by employing run-time loadable and registrable enabling technology plug-ins which do not reference any specific managed data network object types and managed data network entity specification files contain enabling technology specific use directives which are run-time processed by the generic lexical analyzer augmented with enabling-technology-specific lexical analyzer stubs. The directive parser is employed to parse each managed data network entity specification file, and is not an object code parser. It is respectfully submitted that Van de Vanter does not describe a directive parser but rather a character token parser. Van de Vanter does not teach or suggest a generic lexical analyzer run-time augmented with enabling-technology specific lexical analyzer stub processing enabling-technology specific use directives parsed from a managed data network entity specification file using a directive parser.

The framework of amended claim 1, includes an executable code implementation of a single managed entity object class. As recited in claim 1, the single managed entity object class is run-time derivable via type derivation based on run-time parsed entity derivation directives. The Examiner has equated the topology objects of Haggerty with the single managed entity object class of claim 1. The Applicant agrees that in Haggerty multiple topology objects are derived from the managed object of Haggerty. However, Haggerty teaches the use of pre-compiled topology object types as opposed to a single run-time derivable managed entity object

Application No. 10/021,080
Response Dated 02/20/2006
Reply to Office Action of 11/01/2005

PATENT

Agent's Docket No. 11922-US

class. As evidenced from Haggerty's reduction to practice, Haggerty's derivation is performed at compile time as stated at page 77 column 1 lines 1 to 3: "In our implementation, we have defined derivatives of these objects for actual enterprise-specific equipment." The applicant respectfully submits that a person of ordinary skill in the art would not interpret Haggerty as teaching performing run-time type derivation of the managed object. Furthermore, a person of ordinary skill in the art would interpret the cited passage "topology objects are created by ... auto discovery" to mean that compile-time-derived topology objects are instantiated during auto discovery of field installed network elements. Run-time instantiation, which builds a containment hierarchy 500 shown in the present application in Fig. 5, is different from compile-time type derivation employed by Haggerty (see page 77 column 1 lines 1 to 3). Furthermore, run-time instantiation, which builds a containment hierarchy 500 shown in the present application in Fig. 5, is different from run-time type derivation claimed in independent claims 1 and 12 of the present application which build at run-time the single class type derivation hierarchy 300 shown in Fig. 3 of the present application. The applicant therefore respectfully submits that Haggerty does not teach or suggest an executable code implementation of a single managed entity object class which is run-time derivable via type derivation based on run-time parsed entity derivation directives.

The framework of amended claim 1 includes a generic lexical analyzer which is run-time augmented with enabling-technology-plugin-in-specific lexical analyzer stubs. Van de Vanter does not describe a run-time augmentable lexical analyzer, but rather separate pre-compiled and linked-in lexical analyzers. A first lexical analyzer identifies character tokens which signal the deactivation of the first lexical analyzer and the exclusive activation of a second lexical analyzer. This is described in paragraph [30] of the Van de Vanter reference. The applicant therefore

Application No. 10/021,080
Response Dated 02/20/2006
Reply to Office Action of 11/01/2005

PATENT

Agent's Docket No. 11922-US

respectfully submits that Van de Vanter does not teach the generic lexical analyzer which is run-time augmented with enabling-technology-plugin-in-specific lexical analyzer stubs.

The framework of amended claim 1 includes a message interpreter processing, at run-time, messages received from at least one network management and service provisioning software application. The applicant respectfully submits that too much emphasis is put on the vague statement of functionality in the second paragraph of the second column on page 78 "integrates with OpenView" in asserting the teaching of the interpreter element in Haggerty's description. It is respectfully submitted that "the user interfaces launched from OpenView", described by Haggerty, correspond to network management and service provisioning software applications and not to a managed data network object instance. Applicant respectfully submits that in the second paragraph of the second column on page 78 Haggerty describes in very generic terms only results indicating that software applications integrate with the network management solution described by Haggerty, results which are irrelevant to the present invention. As stated above the present invention achieves a separation between managed data network entities and software applications enabling independent development, maintenance and troubleshooting. The message interpreter element of the network management and service provisioning framework provides the necessary run-time interworking by invoking at run-time managed data network object methods in respect of particular managed data network object instances in accordance with messages received at run-time from software applications. The applicant therefore respectfully submits that Haggerty does not teach or suggest the claimed message interpreter.

Furthermore, the collection of all the elements and their interaction must be considered. The elements and characterizations of claim 1 allow a more flexible development of network management and service provisioning system architecture separating data network entities,

Application No. 10/021,080
Response Dated 02/20/2006
Reply to Office Action of 11/01/2005

PATENT

Agent's Docket No. 11922-US

enabling technologies, and software applications. The separation enables independent development, maintenance, and troubleshooting of network management and service provisioning deployments minimizing the need to re-code and re-compile framework code in support of new managed entity object types. The source code of the framework need only be coded and compiled once separate from enabling technology particulars, managed data network entity particulars, and software application particulars. Similarly enabling technology particulars are coded and compiled into plug-ins separately from framework particulars, managed data network entity particulars, and software application particulars. Also, the single manageable entity object class is coded and compiled separately from enabling technology particulars and software application particulars. The software applications are coded and compiled separately from the framework, enabling technology particulars, and managed data network entity particulars. The lexical analyzer stubs are developed separately from managed data network entity particulars and software application particulars. It will be especially noted that the framework object code, the enabling technology plug-in object code, lexical analyzer stub object code, and software application object code are never linked. The notable lack of linking of the object code prior to the deployment of the claimed network management and service provisioning system is compensated for at run-time by: the directive parser which parses directives from managed entity specification files, the managed data network entity specification files which specify how to derive the single managed entity object class, the plug-in registry which registers enabling technology plug-ins, and the augmentable lexical analyzer which processes enabling technology use directives parsed by the directive parser; and the message interpreter which processes software application directives included in messages received from software applications, claimed elements which are not taught by Haggerty.

Application No. 10/021,080
Response Dated 02/20/2006
Reply to Office Action of 11/01/2005

PATENT
Agent's Docket No. 11922-US

In view of the above, a person skilled in the art can only conclude that the solution taught by Haggerty is not a run-time expandable network management and service provisioning system but rather a pre-compiled network management system expandable only through re-coding and re-compilation. Further Van de Vanter does not teach a run-time augmentable lexical analyzer; rather Van de Vanter teaches multiple pre-compiled lexical analyzers.

Haggerty by itself, and in combination with Van de Vanter, fails to teach each and every element of the network management and service provisioning framework of amended claim 1. Independent amended claim 12 includes limitations similar to those of amended claim 1. Amended claims 2 to 3, 6 to 9, 13 to 20 and 22 to 24 are variously dependent on amended independent claims 1 and 12, and include all limitations thereof. For these reasons, the applicant respectfully submits that claims 1 to 3, 6 to 9, 12 to 20, and 22 to 24 of the present application, as amended, are not obvious in view of the combination of Haggerty and Van de Vanter.

In his 04/01/2005 communication, the Examiner rejected objected to claims 4 and 5 under 35 U.S.C. 103, as obvious in view of Haggerty and Acker. Reasons in support of non-obviousness were presented in the response dated 07/29/2005. In his 11/01/2005 communication the Examiner is silent regarding the submitted non-obviousness reasons in respect of claims 4 and 5. It can only be concluded that the Examiner has considered the non-obviousness reasons submitted in the 07/29/2005 response and that the reasons were persuasive. If the reasons were not persuasive then there would not have been a reason for the Examiner to raise the present obviousness objection in respect of claims 4 and 5.

Except for the statement of objection, the Examiner's reasons for objecting to claims 4 and 5 do not mention how the teachings of Van de Vanter are relevant. In fact, the reasons for

Application No. 10/021,080
Response Dated 02/20/2006
Reply to Office Action of 11/01/2005

PATENT
Agent's Docket No. 11922-US

objecting to claims 4 and 5 stated in the 11/01/2005 communication appear to be identical to the reasons for objecting to claims 4 and 5 stated in the 04/01/2005 communication. Applicant respectfully submits that the Examiner did not provide necessary objective evidence of record of any teaching, motivation or suggestion for combining the references to enable an application of Section 103(a) to show which elements of the Van de Vanter combine with Haggerty and Acker to obviate the invention as claimed in claims 4 and 5. It is therefore respectfully submitted that the subject matter claimed in amended claims 4 and 5 is non-obvious.

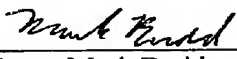
Furthermore, claims 4 and 5 are dependent on claim 1, as explained above, neither Haggerty nor Van de Vanter teach all the elements of claim 1. The applicant submits that these elements are also not taught or suggested by Acker and that a prima facie case of obviousness has therefore not been established against claims 4 and 5.

Reconsideration and allowance are respectfully requested.

Respectfully submitted,

Richard J. Mitchell
Registration No. 34519
Agent of Record

MARKS & CLERK
P. O. Box 957, Station B,
Ottawa, Ontario, Canada
K1P 5S7
(613) 236-9561



Stuart Mark Budd
Registration No. 53,880
Agent of Record

MARKS & CLERK
P. O. Box 957, Station B,
Ottawa, Ontario, Canada
K1P 5S7
(613) 236-9561